The Dual Codebook

Maas, Jonathan L.; Okafor, Emmanuel; Wiering, Marco

Published in:
Belgian-Dutch Artificial Intelligence Conference (BNAIC)

Publication date:
2016

The Dual Codebook: Combining Bags of Visual Words in Image Classification

Jonathan L. Maas [a] Emmanuel Okafor [a] Marco A. Wiering [a]

[a] *Institute of Artificial Intelligence and Cognitive Engineering. University of Groningen, Nijenborgh 9, Groningen, The Netherlands*

Abstract

In this paper, we evaluate the performance of two conventional bag of words approaches, using two basic local feature descriptors, to perform image classification. These approaches are compared to a novel design which combines two bags of visual words, using two different feature descriptors. The system extends earlier work wherein a bag of visual words approach with an L2 support vector machine classifier outperforms several alternatives. The descriptors we test are raw pixel intensities and the Histogram of Oriented Gradients. Using a novel Primal Support Vector Machine as a classifier, we perform image classification on the CIFAR-10 and MNIST datasets. Results show that the dual codebook implementation successfully utilizes the potential contributive information encapsulated by an alternative feature descriptor and increases performance, improving classification by 5-18% on CIFAR-10, and 0.22-1.03% for MNIST compared to the simple bag of words approaches.

1 Introduction

In this paper, we propose and evaluate the use of a Dual Bag Of visual Words model (Dual-BOW) in a relatively conventional framework to perform image classification. Within computer vision, there are many approaches that have been used to create image recognition systems [12]. The challenge which renders classic conventional machine learning techniques inaccurate revolve around representing and encapsulating the essential and unique features of an object or entity, which may occur rotated, scaled, illuminated, or oriented differently.

A popular approach which can encapsulate this is known as the bag of visual words (BOW) [4], which has been shown to reach good performances on multiple tasks [2, 3] and is also simple in design. Recently, improvement with the use of a bag of visual words with local feature descriptors has been applied in domains such as facial recognition, character, animal and object recognition. This is evident in the works of [16] whereby the bag of visual words with the histogram of oriented gradient (HOG-BOW) showed a superior performance relative to other local feature descriptors on different character datasets. Also, the authors in [8] showed that the use of HOG-BOW outperforms several classical based methods such as HOG, SIFT, and a multi-subregion based correlation filter bank (MS-CFB) on a facial dataset (FERET). Though studies by authors in [14] have shown that the combination of several feature descriptors which they called the Joint learning framework outperforms the BOW [14] and HOG-SIFT-BOW [11] approaches.

In this paper, we show the superiority of a bag of visual words with the combination of two local feature descriptors by creating a dual codebook which contains both local features (Dual BOW) compared to the conventional bag of visual words methods (BOW and HOG-BOW) with a single codebook. Our goal of this study is to research the additional effect of combining two bags of words, using different local feature descriptors (LFD). Under the notion that different feature descriptors may encapsulate different essential information, we will assess the performance increase (if any) of combining this information with respect to the conventional bag of visual words, which utilizes only single feature descriptors.

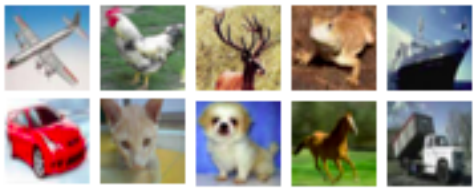

Figure 1: Samples from the CIFAR-10 dataset.

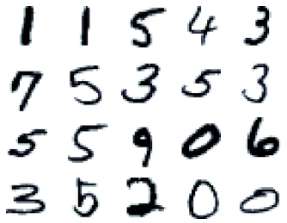

Figure 2: Samples from the MNIST dataset.

Outline This paper is organized as follows: in Section 2, we describe the datasets used in our experiment. Hereafter, we discuss the system design, local feature descriptors, bag of word models, and the used classifier in Section 3. Having covered the basis of our implementation, we will discuss our experiments and results in Section 4. Lastly, we discuss our findings in the conclusion in Section 5.

2 Datasets

For our experiment, we decided to use two datasets to achieve a more reliable assessment of the potential benefit of our proposed approach. Two popular, and diverse, benchmarks datasets often used in this field are the MNIST and CIFAR-10 datasets. MNIST [10] (see Figure 2) consists of 70,000 (60,000 training, 10,000 testing) 28 x 28 pixel images of 10 classes of digits. Though often considered a simplistic dataset, it remains a popular benchmark and provides plenty research to compare with. CIFAR-10 [9] (see Figure 1) consists of 60,000 (50,000 training, 10,000 testing) 32 x 32 colour images, constructed from 10 more diverse classes (ranging from animals to vehicles).

The bag of words approach we work with relies on the extraction of so called patches, sub-parts of the image, that can be extracted using a sliding window of a fixed size. For MNIST, the images were rescaled (using cubic interpolation) to an image resolution of 48 x 48 pixels, after which patches of 14 x 14 pixels were extracted. For CIFAR-10, smaller patchsizes of 8 x 8 were more appropriate, as patch size appeared to have a large impact depending on the dataset used. The image size remained unchanged at 32 x 32 pixels.

3 System Design

The system design builds upon the framework used in [16], wherein a bag of visual words is used, and the performance of several different local feature descriptors was evaluated. Herein, they also compare the performance of several types of support vector machines.

We designed our system with flexibility in mind, as such that it enables swapping different local feature descriptors[1] to be used in combination with bag of word approaches, allowing different patch sizes, and implementation methodologies.

3.1 Conventional Bag of Words

The Bag of Visual Words has been a popular tool in computer vision and classification [4], wherein an image is represented by regarding the patches that it is composed of. Patches are described by an appropriate local feature descriptor, which is used to construct their patch-features. Using this methodology, one can create a bag of words by applying an unsupervised algorithm (such as K-means clustering [13]) on a random collection of patches, extracted from images from the training set. The resulting centroids are intended to represent generalized patches, or visual words, and as a whole act as a dictionary (which we refer to as a codebook within the context of this paper), representing which visual elements are acknowledged to exist and occur in the data [16].

Once the codebook is constructed, it can be used to represent a new image. This is done by partitioning a given image N into S (non-overlapping) segments, of equal size. Within every segment, n

[1] For the experiment, however, only two local feature descriptors were used. We also intended to include a local binary patterns feature descriptor, but at the time did not possess the computational resources to include it in our research.

patches are extracted using a sliding window of a custom size and shift. The derived set of patches are then described by feature vectors using the appropriate local feature descriptor.

Hereafter, the activations are computed in the following fashion. For every patch-feature $p_i \in \mathbb{R}^n$ from the collection of patches within a segment, distances are computed to each word $w_j \in \mathbb{R}^n$ from a codebook $C^l = \{w_1, w_2 ... w_K\}$ (where $l \in \{IMG, HOG, DUAL\}$ denotes the appropriate feature descriptor), using a distance function $d\left(p_i, w_j\right)$. In our experiment, we used the Euclidean distance as distance function:

$$d\left(p_i, w_j\right) = \sqrt{\sum_{x=1}^{n} \left(p_i^x - w_j^x\right)^2} \tag{1}$$

to represent the distance from a patch p from an image to centroid w from the codebook, over all elements of its feature vector length.

Computing the distance to all words allows us to compute the mean distance of patch p_i to all words:

$$\bar{d}(p_i, w) = \frac{\sum_{j=1}^{K} d(p_i, w_j)}{K} \tag{2}$$

We will compute the cluster activations according to the Soft-Assignment function [3], by updating the activation vector $a_j \in \mathbb{R}^K$, which denotes the activations of the codebook centroids with respect to the patches within the segment. For every patch $p_i \in \mathbb{R}^n$, the activation value (a_j) of word w_j is updated by:

$$a_j = \begin{cases} a_j, & \text{if } \delta \leq 0 \\ a_j + \delta, & \text{if } \delta > 0 \end{cases} \tag{3}$$

Where $\delta = \bar{d}(p_i, w) - d(p_i, w_j)$ (and corresponds to a similarity measure between a patch and a word). Repeating this procedure for every patch within segment $s \in \mathbb{R}^S$ gradually generates an activation vector for segment s:

$$A_s(K) = \{a_1, a_2, ... a_K\} \tag{4}$$

To create the final feature vector, $x_{_N}^l$, representing a given image N, using codebook l (and its corresponding local feature descriptor), the activations of all S segments of the image are concatenated:

$$x_{_N}^l(s) = \{A_1; A_2; ... A_S\} \tag{5}$$

and standardised once.

The resulting final feature vector can be used as training and testing data for any classifier of choice. Obviously, computational complexity in this approach grows with feature descriptor size, and the number of centroids used. The dimensionality of the final feature vector of the image, corresponds to $S * K$, where S corresponds to the number of segments the image is partitioned in, and where K is the number of centroids in the codebook used. The codebooks we used in our approach are generated using 200,000 patches randomly extracted from the dataset used, and clustered (using K-means Clustering) using 150 iterations. Having described the bag of words approach, we now describe the methods used in our experiment.

3.1.1 Bag of Visual Words with Pixel Intensities (BOW)

In its most conventional implementation, the bag of visual words approach uses patches described by their raw pixel intensities. The raw pixel intensities method directly uses the RGB intensities of the pixels within a patch. Simple as it may be, its successes in several tasks have shown its potential [15], and show that raw pixel intensities within patches can be used to represent interesting features. Nevertheless, the feature vector length can grow very large when larger patches are used, especially in colour images (which is the case for the 3-channel CIFAR-10 dataset, as opposed to the single-channel MNIST dataset).

In our experiments, for MNIST, the patch size of 14 x 14 pixels results in a patch-feature size of 196 elements. For CIFAR-10, however, we need to track three colour channels of a 8 x 8 pixel patch, which results in a patch-feature length of 192. After computing the patch-feature vector, it is standardised.

Though we included modules for performing different levels of pre- and postprocessing, we settled on using only standardisation where appropriate. Standardisation of a vector is performed by computing the mean of its elements:

$$\overline{x} = \frac{\sum_{i=1}^{n} x_i}{n} \tag{6}$$

Then, the deviation is computed by:

$$\sigma = \sqrt{\frac{\sum_{i=1}^{n} (\overline{x} - x_i)^2}{n} + e} \tag{7}$$

Where e is used as a small constant to avoid a zero standard deviation. Then the standardised vector is obtained by updating the vector values:

$$x'_i = \frac{x_i - \overline{x}}{\sigma} \tag{8}$$

We used this standardisation scheme on several occasions within the design. For our experiment, we ran two configurations of the IMG implementation, one using 400 centroids for its codebook, the other using 800. Images are partitioned into 9 segment (3 x 3), and this results in a final feature dimensionality of 3,600 and 7,200 for the 400 and 800 centroids approach respectively.

3.1.2 Bag of Visual Words with Histogram of Oriented Gradients (HOG-BOW)

An alternative to the raw pixel intensities is to use the Histogram of Oriented Gradients (known as HOG) to describe patches. The Histogram of Oriented Gradients [5] has been a popular feature descriptor for a long while, and knows several different uses [17, 16]. To compute the descriptor, gradient components are computed for the horizontal and vertical gradient (G_x and G_y respectively) for every pixel in the patch. Though multiple masks can be used, the simple kernel $[-1, 0, +1]$ bears preference [1]. The gradients are computed with:

$$G_x = f(x+1, y) - f(x-1, y) \tag{9}$$

$$G_y = f(x, y+1) - f(x, y-1) \tag{10}$$

where $f(x, y)$ is the pixel intensity at coordinate x,y. The final Magnitude $M(x, y)$ (intensity of change) and orientation $\theta(x, y)$ (direction of change) are computed as:

$$M(x, y) = \sqrt{G_x^2 + G_y^2} \tag{11}$$

$$\theta(x, y) = \tan^{-1} \frac{G_y}{G_x} \tag{12}$$

After computing the magnitudes and orientations for every pixel, the patch is segmented into four quadrants.Within each quadrant, the magnitudes of all pixels are binned using linear interpolation (thus the binned magnitude is distributed over the neighbouring bins) into a histogram by the corresponding orientations, which produces the Histogram of Oriented Gradients. After computing the histograms of all four quadrants, these are concatenated to produce the feature vector representing the patch.

For our experiment, we used 9 bins to represent orientations in a range of $0 - 180^o$ (thus a bin width of 20 degrees). Since the patch sizes do not determine the HOG's feature vector size, the feature vector length for MNIST is 36. For the tri-colour channel CIFAR-10, it is 108.

For MNIST, a patch size of 14 x 14 pixels is reduced to 12 x 12 to cope with padding, after which HOG is computed for four 6 x 6 pixel cells. For CIFAR-10, a patch size of 8 x 8 pixels is reduced to 6 x 6 for the same reason, and the HOG is computed for four 3 x 3 pixel cells. As with the raw pixel intensities local feature descriptor, the HOG feature vector is also standardised.

For our experiment, we also ran two configurations of the HOG-BOW implementation, mirrorring the IMG runs with one implementation using 400 centroids for the codebook, the other 800. The final feature dimensionality remains unchanged at 3,600 (for 400 centroids) and 7,200 (for 800 centroids) since the final feature length is unaffected by the length of the local feature descriptor used.

3.2 Dual Bag of Visual Words: combining Pixel Intensity and Histogram of Oriented Gradients (Dual BOW)

We propose the combination of both the raw pixel intensities and HOG features to develop a dual codebook. This enigma of combining features within the scope of the visual bag of words approach knows little prior research [6]. In essence, the dual codebook is the combination of two codebooks, which may have been generated either using the same local feature descriptor (possibly under a different configuration), or an entirely different one. The configuration of the second codebook is not bound by those used in the first, and thus may also operate with a different number of centroids.

In this fashion, given two codebooks C^{IMG} and C^{HOG} (generated using raw pixel intensities, and the histogram of oriented gradients respectively), an image N is represented by computing the activations, x_N^l, for both codebooks towards this image. The activation vectors obtained, x_N^{IMG} and x_N^{HOG} are then concatenated:

$$x_N^{DUAL} = \left\{ x_N^{IMG}; x_N^{HOG} \right\} \tag{13}$$

to create the final feature vector of the image under the dual codebook approach.

This approach effectively allows the combination of two different local feature descriptors, which can aid classification accuracy by the inclusion of potentially essential information which may be encapsulated by the one, but not the other feature descriptor.

In our experiment, the dual codebook was evaluated under the same configurations as its singular alternatives, and combines two codebooks of 400 centroids each. This configuration therefore results in a final feature vector with a dimensionality of 7,200. Based on the dual codebook used in this section, the new bag of visual word formed is referred to as Dual-BOW.

3.3 Classifier

For classification, we designed an L2 'primal' support vector machine (one for each class) using a revised objective function:

$$\min_{\omega,b} L = \|\omega\|^2 + C \cdot \sum_N \xi_N^2 \tag{14}$$

and output function:

$$g(x_N) = \omega \cdot x_N + b \tag{15}$$

where $x_N = x_N^l$ denotes the centroid activations from the bag of words, using descriptor l, and the error is represented as:

$$\xi_N = \max(0, 1 - y_N \cdot g(x_N)) \tag{16}$$

$y_N \in \langle -1, 1 \rangle$ represents whether the target label of example x_N belongs to the class which this SVM represents.

Training is done in iterations, and all training data are presented in each iteration. For every iteration, if the output doesn't perfectly predict the class ($y_N \cdot g(x_N) < 1$), then the weights are adjusted using the formula:

$$\Delta w_j = -\lambda \cdot (\frac{w_j}{C} - (y_N - g(x_N)) \cdot x_N^j) \tag{17}$$

Where λ denotes the learning rate. At the end of every iteration, the bias b is updated to represent the mean error $y_N - g(x_N)$ of all examples where $y_N \cdot g(x_N) < 1$.

We used an L2 primal Support Vector Machine, with a learning rate λ of 0.0000001, and performed 2000 training iterations before testing. The initial weight values are 0.000002, and C is set to 2048.

4 Results

In total, for both MNIST and CIFAR-10, we designed 5 experiment configurations. For the single bag of word approaches (BOW and HOG-BOW) we performed runs with codebooks of 400 and 800 centroids, whereas the dual codebook implementation was run with two codebooks of 400 centroids each.

We performed 10-Monte Carlo cross validation runs for every of the 5 configurations (BOW-400, BOW-800, HOG-BOW-400, HOG-BOW-800, DUAL-2x400). The results are shown in Table 1.

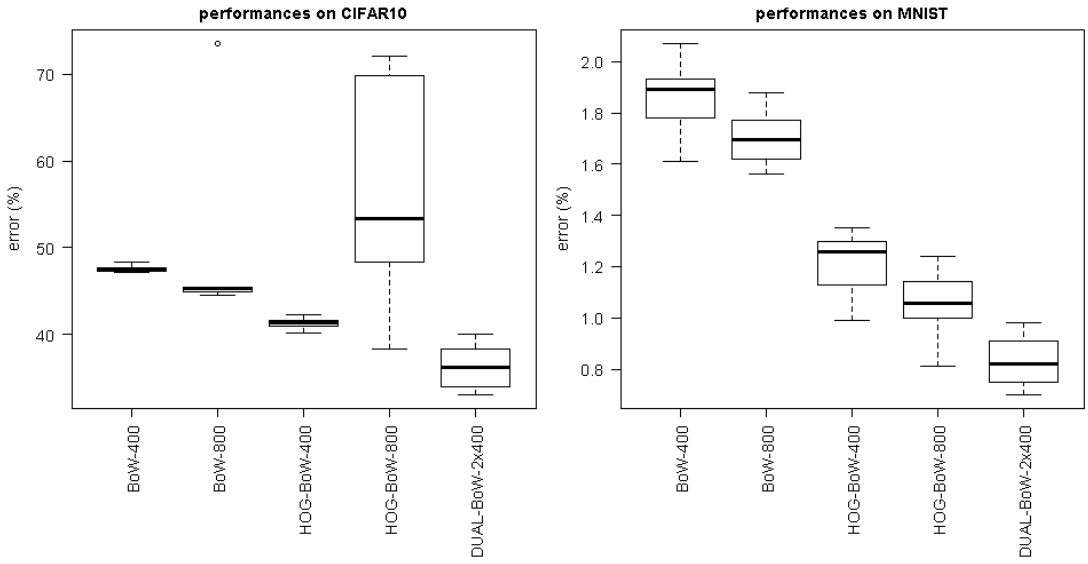

Figure 3: Error rates for CIFAR-10(left) and MNIST(right)

Methods	**MNIST**		**CIFAR-10**	
	Mean	SD	*Mean*	SD
BOW-400	1.85	0.14	47.59	0.42
BOW-800	1.71	0.10	47.96	9.00
HOG-BOW-400	1.22	0.12	41.28	0.61
HOG-BOW-800	1.05	0.13	54.98	12.64
Dual-BOW-2x400	0.83	0.09	36.20	2.60

Table 1: Classification Error (in %) on test-sets of MNIST and CIFAR-10, 10-fold Monte Carlo Cross Validations.

4.1 Evaluation of the CIFAR-10 Dataset

The results of classification on the CIFAR-10 dataset can be seen in Table 1 (and is visualized in Figure 3). As shown, the dual codebook reaches commendable classification performance. Though not stellar nor exceeding present state-of-the-art performance [7], the results still reflect the added value of the dual codebook, resulting in a significant performance increase compared to all single codebook variants.

Student's T-tests shows the dual codebook (Dual-BOW) performs better than the Histogram of Oriented Gradients with 400 centroids ($t = 6.01$, $p < 0.05$), outperforms the 800-centroid variant ($t = 4.60$, $p < 0.05$), and surpasses both 400 and 800-centroid raw pixel intensities (conventional BOW) implementations ($t = 13.26$, $p < 0.05$ and $t = 3.97$, $p < 0.05$, respectively).

Therefore, on CIFAR-10, the Dual-BOW approach, which employs the dual codebook, appears superior to both BOW and HOG-BOW which use only a single codebook, because it obtains the lowest error rate.

4.2 Evaluation of the MNIST Dataset

Though performance improvements may not be as pronounced as those in CIFAR-10, the dual codebook again significantly outperforms all single codebook configurations (see Table 1, and Figure 3).

Student's T-test indicate that the dual codebook approach (Dual-BOW) displays significant improvements over HOG-BOW-400 and HOG-BOW-800 ($t = 8.26$, $p < 0.05$ and $t = 4.50$, $p < 0.05$ respectively). With regard to raw pixel intensities, the Dual-BOW approach significantly outperforms both the BOW-400 ($t = 19.01$, $p < 0.05$) and BOW-800 ($t = 19.97$, $p < 0.05$) implementations.

Thus, the results on MNIST confirm those of CIFAR-10, showing that the Dual-BOW again outperforms conventional BOW approaches utilizing only single codebooks.

4.3 Discussion of Results

In this paper, we have demonstrated the dual codebook's superiority over comparable single codebook approaches, showing a consistent performance improvement over two substantially different datasets. This implies the capability of successfully combining the essential information encapsulated by different local feature descriptors, improving classification performance.

Though both the datasets and the approach used may be considered simplistic by current standards, it does not appear that the dual codebook approach would perform worse with alternative datasets, than single codebook alternatives would.

5 Conclusion

Though performance on either dataset is not present state-of-the-art, it should be kept in mind that many of the data-preprocessing enhancements and excessive parameter tuning conventionally performed for these datasets were not applied, as we intended to study the exclusive benefit of the dual codebook approach, with regard to conventional bag of words approaches that utilize only a single codebook. Therefore, these results say little about the limits of the dual codebook approach, which was used in a quite simple configuration in this experiment. Under slightly more computationally demanding configurations of the primal SVM, performance for CIFAR-10 for the dual codebook reached scores up to 73.18%, and for MNIST up to 99.3%. However, these results were discarded under the need to perform cross validations with limited computational resources, and time constraints.

With regard to future research, there are many possibilities. We intend to expand the design to an N-codebooks implementation, which will be able to combine N bags of words in order to investigate if this can increase performance further.

Additionally, it might be worth investigating the potential value of combining codebooks of the same feature descriptor, but under different configurations (for example, Histogram of Oriented Gradients with a different segmentation grid, or different bin distributions). Other grounds for further research could focus on the necessary sizes of the codebooks in regard to feature vector dimensionality, as it would be ideal if one were able to improve performance by incorporating a mere 100-centroid small extra codebook, which might be based on a local feature descriptor with a computational complexity or intensity too high to consider for larger codebooks.

In regard to the use of the L2 primal support vector machine as classifier, it proved to be more efficient to train than the conventional support vector machine implementation. Though a drawback still remains in an undeniable necessity for parameter optimization. Concerning computational intensity, one might consider the learning rate used (0.0000001) in combination with the number of iterations (2000).

We hope to develop an open framework[2] which combines not only easy modularity and flexibility of combining a number of codebooks, but also remains open to recycling of codebooks, exporting and importing centroids derived from previously trained codebooks, to allow the user to avoid the need to re-train the entire codebook.

References

[1] Jon Arrspide, Luis Salgado, and Massimo Camplani. Image-based on-road vehicle detection using cost-effective histograms of oriented gradients. *Journal of Visual Communication and Image Representation*, 24(7):1182 – 1190, 2013.

[2] A. Coates, B. Carpenter, C. Case, S. Satheesh, B. Suresh, T. Wang, D. J. Wu, and A. Y. Ng. Text detection and character recognition in scene images with unsupervised feature learning. In *Proceedings of the 2011 International Conference on Document Analysis and Recognition*, pages 440–445, 2011.

[2]The framework is currently available online at https://github.com/JonathanMaas/nCodebooks

[3] A. Coates, H. Lee, and A.Y. Ng. An analysis of single-layer networks in unsupervised feature learning. In Geoffrey Gordon, David Dunson, and Miroslav Dudk, editors, *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, volume 15 of *JMLR Workshop and Conference Proceedings*, pages 215–223. JMLR W&CP, 2011.

[4] G. Csurka, C. Bray, C. Dance, and L. Fan. Visual categorization with bags of keypoints. *Workshop on Statistical Learning in Computer Vision, ECCV*, pages 1–22, 2004.

[5] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 886–893 vol. 1, 2005.

[6] Huilin Gao, Wenjie Chen, and Lihua Dou. Image classification based on support vector machine and the fusion of complementary features. *CoRR*, abs/1511.01706, 2015.

[7] Benjamin Graham. Fractional max-pooling. *CoRR*, abs/1412.6071, 2014.

[8] Mahir Faik Karaaba, Olarik Surinta, L. R. B. Schomaker, and Marco A. Wiering. Robust face identification with small sample sizes using bag of words and histogram of oriented gradients. In *Proceedings of the 11th Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*, pages 582–589, 2016.

[9] A. Krizhevsky and G. Hinton. Learning multiple layers of features from tiny images. *Master's thesis, Department of Computer Science, University of Toronto*, 2009.

[10] C. LeCun, Y. Cortes. The mnist database of handwritten digits. 1998.

[11] Kart-Leong Lim and Hamed Kiani Galoogahi. Shape classification using local and global features. In *Image and Video Technology (PSIVT), Fourth Pacific-Rim Symposium on*, pages 115–120, 2010.

[12] D. Lu and Q. Weng. A survey of image classification methods and techniques for improving classification performance. *International Journal of Remote Sensing*, 28(5):823–870, 2007.

[13] J. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics*, pages 281–297, Berkeley, Calif., 1967. University of California Press.

[14] Bharath Ramesh, Cheng Xiang, and Tong Heng Lee. Shape classification using invariant features and contextual information in the bag-of-words model. *Pattern Recognition*, 48(3):894–906, 2015.

[15] O. Surinta, L. Schomaker, and M. Wiering. A comparison of feature and pixel-based methods for recognizing handwritten Bangla digits. In *12th International Conference on Document Analysis and Recognition*, pages 165–169, 2013.

[16] Olarik Surinta, Mahir F. Karaaba, Tusar K. Mishra, Lambert R. B. Schomaker, and Marco A. Wiering. *Recognizing Handwritten Characters with Local Descriptors and Bags of Visual Words*, pages 255–264. Springer International Publishing, Cham, 2015.

[17] Kazuhiko Takahashi, Sae Takahashi, Yunduan Cui, and Masafumi Hashimoto. *Remarks on Computational Facial Expression Recognition from HOG Features Using Quaternion Multi-layer Neural Network*, pages 15–24. Springer International Publishing, Cham, 2014.